Speaking "Data Maintenance"An intro to Data-related presales

Glossary of Partners

Mulesoft

Salesforce-owned company.

Salesforce will tell you about Mulesoft, and forget to tell you if they’re speaking about Anypoint, Composer, or anything else about Mulesoft. Make sure they are targeting the discussion in ways that serve the client see data volumes, mappings, complexity) rather than the product.

Talend

The Free version is limited and generally is just used to cross load CSV files. The paid version is very powerful but requires an IT team to wield properly, and has setup costs for us regarding how to set the environment in place.

Jitterbit

Is bad. Run away. It used to be the king, but lack of updates, bad infrastructure and bad support lead to it losing ground over the last years.

Boomi

A paid ETL by Dell. Powerful, used by US corporations, but paid. Rarely seen in the wild unless the client already has a license.

Informatica

A paid ETL by Informatica. Powerful, used by US corporations, but paid. Extremely rarely seen in the wild unless the client already has a license.

Kafka

An event bus by Apache. Used by Event Driven Systems

Glossary of Technologies

API

An Application Programming Interface (API) is a set of functions, procedures, methods or classes used by computer programs to request services from the operating system, software libraries or any other service providers running on the computer. A computer programmer uses the API to make application programs.

MDM

Master data management[1] (MDM) is a technology-enabled discipline in which business and information technology work together to ensure the uniformity, accuracy, stewardship, semantic consistency and accountability of the enterprise's official shared master data assets.[2][3]

In simpler terms, it is the act of defining which system has the correct data, where, when, and how it is kept up to date.

Clients often request an “MDM”, which actually just means a “centralized system of record”, meaning a database where they know the data is correct and should always prime in case of data differences with other systems

ETL (Extract-Transform-Load)

A software tool that extracts data from a source system, transforms the data (using rules, lookup tables, and other functionality) to convert it to the desired state, and then loads (writes) the data to a target database.

Web service

A Web service is defined as "a software system designed to support interoperable machine-to-machine interaction over a network". Web services are frequently just Web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services.

REST

Representational state transfer (REST) is a software architectural that was made to guide the development of the World Wide Web. Systems which implement REST are called 'RESTful' systems. REST documents a way for computer systems to communicate with each other using HTTP requests.
It is supported by most recent players, is flexible and cheap.

It is also less secure than SOAP by design, and for high volumes, Events can be better suited.

SOAP

SOAP is a protocol used in computing. Web services use this protocol to communicate. SOAP uses XML to encode a message. It uses other application-layer protocols, for transport, and content negotiation, for example HTTP and Remote procedure call.

It is less flexible than REST and harder to implement, but it offers more security and some calls are specific to SOAP.

GraphQL

An API type that’s similar to REST but has technical differences in implementation and scope of data recovery. Great if multiple calls need to be done of varying scopes on the same endpoint.

Web socket

Much like REST, it is an HTTP API protocol. It has way less flexibility but is great if you want to “just push a message somewhere”, if that message corresponds to a very specific format.

Events

Events operate on the opposite of REST/SOAP calls. In REST/SOAP you tell a system what you want it to do, and add information needed for the action. Events just say “something happened, here’s the data about that”. It becomes the receiving system’s job to interpret the action to do.

Events are asynchronous, and by nature harder to manipulate and ensure than REST/SOAP calls. It’s great for high-volume, low-latency situations, but expensive.

ESB (Enterprise Service Bus)

REST and SOAP historically integrate two different platforms directly. These platforms become “coupled” - if one changes, the other must change to allow the integration to continue.

An Enterprise Service Bus is a platform that sits in the middle of these integrations. All platforms speak to the ESB, and the ESB then manipulates data, streams, events, and whatever else is necessary to allow the platforms to get the information they need back.

Setting up an ESB is costly, and generally leads to restructures in existing integrations so they leverage the new ESB. It does however lower the cost of future integrations, and lowers platform coupling.

It is a good idea to implement an ESB when you have at least 4 platforms speaking together, and it can be valuable to look at it for lower numbers.

Batch

The default Data Loading mode for Data Loader and REST calls.
Accepts Data passed via REST, in batches. Processes these batches *synchronously* and then returns the results as a response with the same number of records as in the original batch, with a status code.

The default batch size in Data Loader is 200. The number of batches submitted for a data manipulation operation (insert, update, delete, etc) depends on the number of records and batch size selected.
One API call is used per batch, which can lead to limit issues for big loads.

Bulk

A different Data Loading mode, usable via Data Loader or REST calls.

Accepts Data passed as a CSV file which must be sent to the server in a series of REST calls. Once all the data has been received, a final call tells the bulk to start. It then processes these batches *asynchronously* and returns the results to the batch, which must then be downloaded via REST calls.

The default BULK size in Data Loader is 2000. The amount of records loadable is by nature very high (a few million), and as such this API is recommended for big data transfers.

Event-Driven Architecture

A situation where the client already uses Event-based systems and expects you to implement a receiving Event Bus and get Events for integrations. See Events.

Database

Often conflated with Relational Database Management System, actually just means a place where data is stored. Can be relational, graph based, events based, whatever. If “database” is said, try to see which kind.

Data Warehouse

Often conflated for “lots of tables”. Actually means place where data frim multiple systems are stored. Doesn’t have to mean that the data is transformed to serve an MDM - you can just store multiple systems and call it a day.

Data Lake

Often conflated for “lots of tables”. Actually has nothing to do with tables, and defines an architecture for data storage, with a heavy focus on data “flatness”, hence “lake”. The data can be structured, semi-structured, unstructured - meaning a Data Analysis team will be needed to use it properly.

If the client is misusing this term, it’s fine. If they’re using it correctly, the complexity of your project just went up.

Data Archival

Taking data from a system and storing it in another when it’s no longer useful but you don’t want to lose it. Generally done for Cost considerations - storing in a local postgredb is cheap.

Glossary of Volumes

Data Storage

Amount of records salesforce stores. Records in Salesforce are generally (exceptions apply to tasks, events, email messages) abstracted to 2kb per record. Storage is expensive in Salesforce, keeping it reasonable generally lowers project cost.

File Storage

Amount of ContentDocument Salesforce stores. Very expensive, and Salesforce does document management poorly. You might want to look into third party solutions.

LDV (Large Data Volumes)

Above 500000 rows in a single table, LDV applies. This is a key word for Architects that will understand they need to watch out for volumes, flows, api calls, storage over time etc.

Revision #2
Created 2022-07-21 10:43:01 UTC by Windyo
Updated 2022-08-29 03:41:53 UTC by thejamesjames